AMENDMENTS TO THE CLAIMS

Please cancel claim 7 without prejudice. Kindly amend claims 1, 6, 8-9, 11, 16-18, 21, and 26-27 as shown in the following listing of claims. The listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims

1. (Currently Amended) A microprocessor apparatus, for precluding a pipeline stall due to microcode ROM access delay, the microprocessor apparatus comprising:

   a translator, configured to generate a plurality of micro instruction queue entries within a micro instruction queue, each of said plurality of micro instruction queue entries corresponding to an instruction, and said each of said plurality of micro instruction queue entries comprising a plurality of micro instructions and a microcode entry point; and

   early access logic, coupled to said micro instruction queue, configured to employ said microcode entry point to access a microcode ROM prior to when said microcode entry point is provided to register logic, whereby said microcode ROM provides a first micro instruction to said register logic when said first micro instruction is required by said register logic.

2. (Original) The microprocessor apparatus as recited in claim 1, wherein said each of said plurality of micro instruction queue entries is provided in order to said register logic.

3. (Original) The microprocessor apparatus as recited in claim 2, wherein said plurality of micro instruction queue entries comprises four micro instruction queue entries.

4. (Original) The microprocessor apparatus as recited in claim 1, wherein said plurality of micro instructions comprises three micro instructions.

5. (Original) The microprocessor apparatus as recited in claim 4, wherein the microcode ROM access delay comprises four clock cycles.

6. (Currently Amended) The microprocessor apparatus as recited in claim 1, wherein said early access logic employs said microcode entry point when said microcode entry point is within a bottom micro instruction queue entry, said bottom micro instruction queue entry comprising one of said each of said plurality of micro instruction queue entries, and wherein said bottom micro instruction queue entry will be provided to said register logic during a next clock cycle.

7. (Cancelled)

8. (Currently Amended) The microprocessor apparatus as recited ~~in claim 7~~in claim 1, wherein said translator is configured to provide a generated micro instruction queue entry to a top micro instruction queue entry, wherein said top micro instruction queue entry comprises one of said each of said plurality of micro instruction queue entries.

9. (Currently Amended) The microprocessor apparatus as recited ~~in claim 7~~in claim 1, wherein said translator is configured to provide a generated micro instruction queue entry to a mux, and, when said plurality of micro instruction queue entries is empty, said mux provides said generated micro instruction queue entry to said register logic during a next clock cycle.

10. (Original) The microprocessor apparatus as recited in claim 9, wherein said early access logic employs a bypass microcode entry point corresponding to said generated micro instruction queue entry.

11. (Currently Amended) An apparatus for absorbing pipeline stalls associated with microcode ROM access delay, the apparatus comprising:

a micro instruction queue, for receiving a plurality of queue entries from a translator, and for providing ~~a plurality~~said plurality of queue entries to register logic, each of said plurality of queue entries comprising:

first micro instructions, all of said first micro instructions corresponding to an instruction; and

a microcode entry point, coupled to said first micro instructions, configured to point to second micro instructions stored within a microcode ROM; and

early access logic, coupled to said micro instruction queue, configured to employ said microcode entry point to access said microcode ROM prior to when said each of said plurality of queue entries is provided to said register logic, whereby a first one of said second micro instructions is provided to said register logic when said first one of said second micro instructions is required by said register logic.

12. (Original) The apparatus as recited in claim 11, wherein said each of said plurality of queue entries is provided in order to said register logic.

13. (Original) The apparatus as recited in claim 12, wherein said plurality of queue entries comprises four queue entries.

14. (Original) The apparatus as recited in claim 11, wherein said first micro instructions comprise three micro instructions.

15. (Original) The apparatus as recited in claim 14, wherein the microcode ROM access delay comprises four clock cycles.

16. (Currently Amended) The apparatus as recited in claim 11, wherein said early access logic employs said microcode entry point when said microcode entry point is within a bottom queue entry, said bottom queue entry comprising one of said each of said plurality of micro instruction queue entries, and wherein said bottom queue entry will be provided to said register logic during a next clock cycle.

17. (Currently Amended) The apparatus as recited in claim 11, ~~further comprising~~:wherein said translator is configured to generate said each

~~a translator, coupled to said plurality of queue entries, configured to generate said each~~ of said plurality of micro instruction queue entries.

18. (Currently Amended) The apparatus as recited in claim 17, wherein said translator is configured to provide a generated queue entry to a top queue entry, wherein said top queue entry comprises one of said each of said plurality of micro instruction queue entries.

19. (Original) The apparatus as recited in claim 17, wherein said translator is configured to provide a generated queue entry to a mux, and, when said plurality of queue entries is empty, said mux provides said generated queue entry to said register logic during a next clock cycle.

20. (Original) The apparatus as recited in claim 19, wherein said early access logic employs a bypass microcode entry point corresponding to said generated micro instruction queue entry.

21. (Currently Amended) A method for precluding microprocessor pipeline stalls resulting from microcode ROM access delay, the method comprising:

translating instructions and providing a corresponding each of a plurality of micro instruction queue entries;

obtaining a microcode entry point from within one of ~~a plurality~~said plurality of micro instruction queue entries, the one of the plurality of micro instruction queue entries comprising first micro instructions; and

employing the microcode entry point to access second micro instructions within a microcode ROM, wherein said employing is performed prior to when the one of the plurality of micro instruction queue entries is routed to a following pipeline stage, and whereby said employing enables the second micro instructions to be provided to the following pipeline stage without incurring the microprocessor pipeline stalls.

22. (Original) The method as recited in claim 21, further comprising:

issuing the plurality of micro instruction queue entries in order to the following pipeline stage.

23. (Original) The method as recited in claim 22, wherein the plurality of micro instruction queue entries comprises four micro instruction queue entries.

24. (Original) The method as recited in claim 21, wherein the first micro instructions comprises three micro instructions.

25. (Original) The method as recited in claim 24, wherein the microcode ROM access delay comprises four clock cycles.

26. (Original) The method as recited in claim 21, wherein said obtaining comprises:

~~Selecting~~ selecting the microcode entry point from within a bottom micro instruction queue entry, the bottom micro instruction queue entry comprising the one of a plurality of micro instruction queue entries, wherein the bottom micro instruction queue entry will be provided to the following pipeline stage during a next clock cycle.

27. (Original) The method as recited in claim 21, further comprising:

generating a current microcode entry point as part of said translating.

28. (Original) The method as recited in claim 27, wherein said generating provides the current microcode entry point to a top micro instruction queue entry, and wherein the top micro instruction queue entry comprises another of the plurality of micro instruction queue entries.

29. (Original) The method as recited in claim 27, wherein said generating provides the current microcode entry point to a mux, and, when the plurality of micro instruction queue entries is empty, the mux provides the current microcode entry point to the following pipeline stage during a next clock cycle.